

Amendments to the Claims:

1 (Currently amended): A computer-implemented method for testing user interface functionality through implicit user interface automation synchronization, the method comprising:

displaying a user interface window on a display of a computer, wherein the user interface window permits action messages that cause a change to the user interface window through asynchronous window procedures, wherein the user-interface window is managed by a thread that processes the action messages, wherein the thread includes a message queue that indicates the current execution of action messages currently being processed by the thread;

hooking the message queue of the thread with a synchronization API of event driven test automation code to bypass the asynchronous window procedures for any action messages received by the thread;

setting a timer to monitor the message queue of the thread hooked with the synchronization API of the event driven test automation code, wherein the timer is configured to monitor the message queue of the thread to determine when the message queue is empty;

upon receiving an action message that causes the change to the user interface window,

determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message, and

when the message queue is empty to indicate completion of the action message, sending a notification via the hooked synchronization API of the event driven test automation code to automatically cause the processing of another action message on the user interface window being displayed on the display of the computer.

~~sub-classing selected windows displayed on the user interface, wherein each of the selected windows corresponds to a message queue of a thread;~~

~~setting timers corresponding to each of the selected windows;~~

~~determining whether a particular timer associate with one of the selected windows has fired; and~~

~~determining whether the message queues of the threads are empty, such that when the message queues are empty, the user interface automation proceeds to a next action.~~

2-22 (Cancelled).

23 (New): The computer-implemented method of claim 1, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message includes calling a wait API

24 (New): The computer-implemented method of claim 1, wherein the synchronization API is stored in a dynamic link library that is loaded into an application that generates the user interface window.

25 (New): The computer-implemented method of claim 24, wherein the synchronization API is implemented by reference to hook code via the dynamic link library.

26 (New): The computer-implemented method of claim 1, wherein the timer monitors other threads beside the thread that includes the message queue that indicates the current execution of action messages currently being processed by the thread.

27 (New): The computer-implemented method of claim 26, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message further includes determining whether a message queue of other threads is empty.

28 (New): The computer-implemented method of claim 1, wherein the message queue is related to system activities related to the user interface window.

29 (New): A computer-readable storage medium having computer executable instructions for testing user interface functionality through implicit user interface automation synchronization, the instructions comprising:

displaying a user interface window on a display of a computer, wherein the user interface window permits action messages that cause a change to the user interface window through asynchronous window procedures, wherein the user-interface window is managed by a

thread that processes the action messages, wherein the thread includes a message queue that indicates the current execution of action messages currently being processed by the thread;

hooking the message queue of the thread with a synchronization API of event driven test automation code to bypass the asynchronous window procedures for any action messages received by the thread;

setting a timer to monitor the message queue of the thread hooked with the synchronization API of the event driven test automation code, wherein the timer is configured to monitor the message queue of the thread to determine when the message queue is empty;

upon receiving an action message that causes the change to the user interface window,

determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message, and

when the message queue is empty to indicate completion of the action message, sending a notification via the hooked synchronization API of the event driven test automation code to automatically cause the processing of another action message on the user interface window being displayed on the display of the computer.

30 (New): The computer-readable storage medium of claim 29, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message includes calling a wait API

31 (New): The computer-readable storage medium of claim 29, wherein the synchronization API is stored in a dynamic link library that is loaded into an application that generates the user interface window.

32 (New): The computer-readable storage medium of claim 31, wherein the synchronization API is implemented by reference to hook code via the dynamic link library.

33 (New): The computer-readable storage medium of claim 29, wherein the timer monitors other threads beside the thread that includes the message queue that indicates the current execution of action messages currently being processed by the thread.

34 (New): The computer-readable storage medium of claim 33, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message further includes determining whether a message queue of other threads is empty.

35 (New): The computer-readable storage medium of claim 29, wherein the message queue is related to system activities related to the user interface window.

36 (New): A system for testing user interface functionality through implicit user interface automation synchronization, the system comprising:

a processor; and

a memory having computer executable instruction stored thereon, wherein the computer executable instructions are configured for:

displaying a user interface window on a display of a computer, wherein the user interface window permits action messages that cause a change to the user interface window through asynchronous window procedures, wherein the user-interface window is managed by a thread that processes the action messages, wherein the thread includes a message queue that indicates the current execution of action messages currently being processed by the thread;

hooking the message queue of the thread with a synchronization API of event driven test automation code to bypass the asynchronous window procedures for any action messages received by the thread;

setting a timer to monitor the message queue of the thread hooked with the synchronization API of the event driven test automation code, wherein the timer is configured to monitor the message queue of the thread to determine when the message queue is empty;

upon receiving an action message that causes the change to the user interface window,

determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message, and

when the message queue is empty to indicate completion of the action message, sending a notification via the hooked synchronization API of the event driven test automation code to automatically cause the processing of another action message on the user interface window being displayed on the display of the computer.

37 (New): The system of claim 36, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message includes calling a wait API

38 (New): The system of claim 36, wherein the synchronization API is stored in a dynamic link library that is loaded into an application that generates the user interface window.

39 (New): The system of claim 38, wherein the synchronization API is implemented by reference to hook code via the dynamic link library.

40 (New): The system of claim 36, wherein the timer monitors other threads beside the thread that includes the message queue that indicates the current execution of action messages currently being processed by the thread.

41 (New): The system of claim 40, wherein determining, by the timer, when the message queue of the thread is empty which indicates completion of the action message further includes determining whether a message queue of other threads is empty.

42 (New): The system of claim 36, wherein the message queue is related to system activities related to the user interface window.